The ACOP Family of Beans: A Framework Independent Approach

Jaka Bobnar, Igor Kriznar (Cosylab, Ljubljana), Piotr Karol Bartkiewicz, Philip Duval, Wu Hong Gong (DESY, Hamburg)

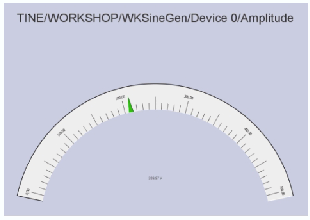
The current ACOP (Advanced Component Oriented Programming) controls set has now been expanded to include a wide variety of graphical java beans, which simultaneously act as displayers of control system data. Besides the original ACOP Chart, the set of ACOP beans also includes a Label, Slider, Table, Gauge, Wheel, and image control, along with an invisible Transport bean, which is itself embedded in the ACOP GUI beans. The new ACOP beans all offer design-time browsing of the control system to expedite data end-point selection. Optionally a developer can choose to connect and render the incoming data automatically, obviating the need for writing code. The developer can either forgo this option or choose to override the generated code with his own, allowing for rich client development. At the same time a user can browse and add or change the control system endpoints at run-time. If the application is using the Component Object Manager (COMA) then all visual aspects of the application can be edited at run-time, allowing for simple client development. This scenario is independent of a framework, and the developer is free to choose the IDE of choice.

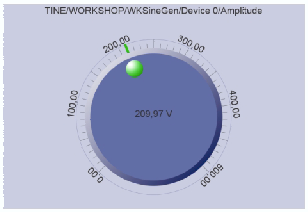
ACOP Displayers

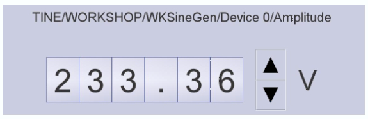
All displayers follow Java Beans conventions.

Single channel displayers:

- AcopDialKnob
- AcopGauger
- AcopIcon
- AcopNumberDisplayer
- AcopSlider
- AcopWheelswitch

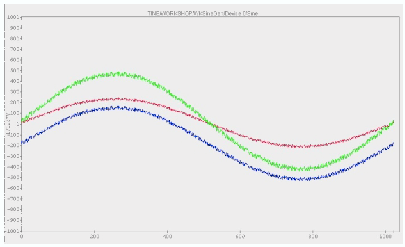


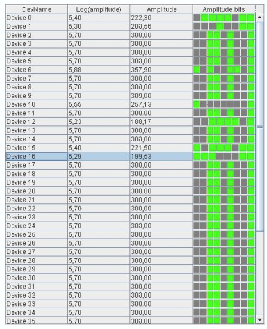




Multiple channel displayers:

- AcopChart
- AcopTable



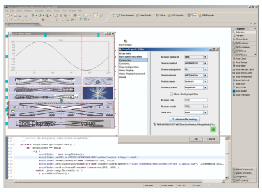


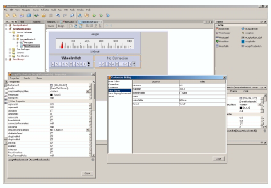
ACOP Transport Bean

- Narrow interface dealing with links (synchronous and asynchronous).
- Each link is specified by ConnectionParameters.
- AcopTransport bean is responsible for all data acquisition required by displayers.
- One AcopTransport can handle a single connection to the server, but can provide data to multiple displayers.
- Can be used as a standalone provider of live data for custom listeners.

GUI Builders

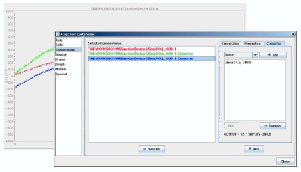
- Rapid application development (RAD)
- Simple clients can be constructed using any Java GUI builder.
- AcopDisplayers and other java beans can be customized using the framework independent customizers and property sheets.
- Not a single line of code is needed to create a simple application.
- Reach client programming still enables full control over generated code and data manipulation.

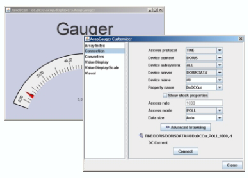




Beans Customization

- All beans implement Java Beans Customizer.
- Beans are fully customizable in design-time and in run-time.
- During design-time customization, java code is generated automatically.
- In design-time beans can be customized using the property sheets provided by Visual Editors.
- Customizers allow setting displayer specific properties (min, max, scale etc.) as well as setting up connection to AcopTransport bean.





Applications

- All ACOP displayers support Drag and Drop. Drag event contains all connection endpoint information as well as display settings.
- Data can be dropped into any application supporting Drag and Drop eg. History Viewer, Notepad.
- Using Container Object MAnager (COMA) applications can be modified in run-time: components can be repositioned, ne components can be added etc. All applied changes are stored into XML configuration file which can be loaded at startup of the application.